Industrial Data
ISSN: 1560-9146
industrial.data@industrial.unmsm.pe
Universidad Nacional Mayor de San Marcos
Perú

Mamani Rodríguez, Zoraida; Del Pino Rodríguez, Luz; Cortez Vasquez, Augusto
Minería de datos distribuida usando clustering k-means en la predictibilidad del proceso petitorio en una organización pública
Industrial Data, vol. 20, núm. 2, 2017, pp. 123-129
Universidad Nacional Mayor de San Marcos
Lima, Perú

Revista Industrial Data 20(2): 123-130 (2017)
DOI: http://dx.doi.org/10.15381/idata.v20i2.13949

ISSN: 1560-9146 (Impreso) / ISSN: 1810-9993 (Electrónico)
Facultad de Ingeniería Industrial - UNMSM

Minería de datos distribuida usando clustering k-means en la predictibilidad del proceso petitorio en una organización pública

RECIBIDO: 11/01/2017 ACEPTADO: 18/05/2017

ZORAIDA MAMANI RODRÍGUEZ[1]
LUZ DEL PINO RODRÍGUEZ[2]
AUGUSTO CORTEZ VASQUEZ[3]

RESUMEN

La minería de datos distribuida está contemplada en el campo de la investigación e implica la aplicación del proceso de extracción de conocimiento sobre grandes volúmenes de información almacenados en bases de datos distribuidas. Las organizaciones modernas requieren de herramientas que realicen tareas de predicción, pronósticos, clasificación entre otros y en línea, sobre sus bases de datos que se ubican en diferentes nodos interconectados a través de internet, de manera que les permita mejorar la calidad de sus servicios. El Clustering es una de las principales tecnicas de modelado de la mineria de datos la cual consiste en dividir la información en grupos diferentes, internamente los miembros de cada grupo son muy similares unos de otros y disimiles respecto a los miembos de los otros grupos. Los grupos o clusters resultantes permiten predecir patrones de comportamiento que pueden aportar en la toma de decisiones de las organizaciones. Es en este contexto que el presente trabajo elabora una propuesta de un prototipo de aplicación de minería de datos distribuida basado en la técnica k-means en la predictibilidad del proceso petitorio de una organizacion pública.

Palabras clave: Minería de Datos Distribuida, Algoritmo Clustering, K-means, petitorio

APPLICATION OF THE DISTRIBUTED DATA MINING USING CLUSTERING K-MEANS IN THE PREDICTABILITY OF THE REQUEST PROCESS OF A PUBLIC ORGANIZATION

ABSTRACT

Distributed data mining is contemplated in the field of research and involves the application of the process of extracting knowledge about large volumes of information stored in distributed databases. Modern organizations require tools that perform tasks of prediction, forecasting, classification and others, online, on their databases that are located in different nodes interconnected through the Internet, in a way that allows them to improve the quality of their services. Clustering is one of the main modeling techniques of data mining which consists of dividing the information into different groups, internally the members of each group are very similar to each other and dissimilar to the members of the other groups. The resulting clusters or clusters allow us to predict patterns of behavior that can contribute to organizational decision-making. It is in this context that the present work elaborates a proposal of a prototype of application of distributed data mining based on the k-means technique in the predictibilidad of the request process of a public organization.

Keywords: Distributed Data Mining, Clustering Algorithm, K-means, Petition

1. INTRODUCCIÓN

La minería de datos distribuida es una disciplina de alto interés de los investigadores debido a las limitaciones que ofrece la mineria de datos centralizada a las realidades organizacionales actuales. Las organizaciones competitivas deben mantenerse dispuestas al cambio, a la mejora continua de los servicios que brindan, respetando los estándares de la industria. Por ello están interesadas en utilizar herramientas informáticas que apoyen sus objetivos.

El clustering es una de las principales tecnicas de modelado de la minería de datos la cual consiste en dividir la información en grupos diferentes, internamente los miembros de cada grupo son muy similares unos de otros y disimiles respecto a los miembos de los otros grupos. Los grupos o clusters pueden ser usados para clasificar nuevos datos (Hurtado, 2005).

En Rekha y Sabu (2010) se define tres modelos de arquitectura de mineria de datos distribuida; el primero consiste en que cada nodo distribuido dispone de un componente de minería encargado de minar los datos en la base de datos local, obteniéndose de esta forma, un modelo de minería de datos parcial en cada uno de los nodos; posteriormente, estos modelos parciales se combinan para obtener el modelo de minería de datos global. Los otros dos modelos son similares pues ambas consideran implementar un modelo global de minería de datos en la parte superior del sistema distribuido que actúe sobre una vista integrada de las distintas bases de datos locales. La diferencia entre estos dos modelos radica en la forma en que se genera la vista integrada sobre la que actuaría la capa de minería de datos. La primera realiza consultas en cada base de datos distribuida de manera independiente, generando un modelo de datos integrado sobre el que operan los algoritmos de minería de datos; mientras que la segunda integra todas las bases de datos distribuidas y las consultas se realizan sobre esta vista integrada de datos.

El organismo judicial peruano ha sido considerado en el presente estudio por tratarse de un modelo de organización moderna cuyos objetivos se encuentran establecidos en la agenda estratégica (Mendoza, 2013); documento que a su vez se encuentra alineado al Plan de desarrollo Institucional al 2018, Plan Bicentenario y al Plan Nacional para la Reforma Integral de la Administración de Justicia.

1 Docente Asociada de la Facultad de Ingeniería de Sistemas e Informatica de la UNMSM. E-mail: zmamanir@unmsm.edu.pe ORCID: 0000-0002-2590-8387
2 Docente Asociada de la Facultad de Ingeniería de Sistemas e Informatica de la UNMSM. E-mail: ldelpinor@unmsm.edu.pe ORCID: 0000-0002-2893-8047
3 Docente Principal de la Facultad de Ingeniería de Sistemas e Informatica de la UNMSM. E-mail: acortezv@unmsm.edu.pe ORCID: 0000-0002-3752-4321

Esta agenda está enfocada en tres ejes: el eje ciudadano, el eje interno y el eje externo. Asimismo, señala como uno de los objetivos principales la predictibilidad de las decisiones judiciales a nivel del eje ciudadano. Además, a nivel del eje interno se precisa como objetivos primordiales la gestión de la calidad jurisdiccional, el uso de las tecnologías de información y justicia electrónica con monitoreo permanente, el uso y análisis de la información estadística y de los indicadores de producción y de calidad.

La reducción de los tiempos procesales permitirá la reducción de la carga procesal del sector judicial; este es otro de los objetivos de la institución; para lo cual se viene realizando estudios de los procesos judiciales que presentan mayor carga procesal. En la Tabla N°1 se expone cuatro de los diez procesos identificados y de todas las especialidades en seis sedes judiciales. Asimismo Breña (2008) realiza un analisis completo de la carga procesal como barrera para el acceso a la justicia.

Tabla 1: *Procesos con mayor carga procesal*

N°	Procesos	Tiempo de demora al 80% (dias)	Motivos de demora que afectan la sentencia
1	Alimentos	100-1200	1.- Oficios a organismos donde labora el demandado y a otras entidades 2.- Escritos relativos a la solicitud y reformulacion de liquidaciones 3.- Solicitud de desarchivamiento 4.- Notificaciones
2	Violencia Familiar	300-600	1.- Oficios relativos a informes psicologicos 2.- Notificaciones
3	Desalojo	600-900	1.- Escrito para impulsar el proceso 2.- Notificaciones
4	Obligaciones de dar sumas de dinero	900-1200	1.- Incumplimiento de requerimiento del juzgado 2.- Escritos para impulsar el proceso 3.- Resoluciones para conocimiento de las partes

Fuente: Proyecto de Mejoramiento de los Servicios de Justicia, 2014

El proceso de negocio sobre el cual se centrará el presente estudio es el petitorio. Veramendi (2008) define el petitorio como lo que se pide sea reconocido o declarado en la sentencia a favor del demandante.

"El petitum es el elemento fundamental de la pretensión del actor en relación con la congruencia de la sentencia ya que ni su objeto inmediato ni mediato puede modificarse a lo largo del proceso ni en la resolución judicial. En pocas palabras, la sentencia debe inexcusablemente ser congruente con la petición." Ezquiaga, 2000: 53 y Sendra, 2007: 209-210 (citado en Veramendi, 2008).

Es en este contexto que se plantea la presente investigación la cual desarrolla un prototipo que aplica minería de datos distribuida sobre datos nominales para determinar patrones de comportamiento en el petitorio de la carga procesal de los periodos 2008 al 2010 correspondiente a un organo jurisdiccional casatorio.

2. SOLUCIÓN PROPUESTA

La metodología de desarrollo que se aplica para la implementación del prototipo de de Minería de Datos Distribuida (MDD) consiste en cinco fases:

A. Diseño del Modelo Dimensional
B. Diseño del Algoritmo MDD
C. Arquitectura del Prototipo
D. Diseño e Implementación del Prototipo
E. Determinación de Resultados

2.1 Diseño del Modelo Dimensional

La arquitectura de datos según la normalización de sus dimensiones es un esquema copo de nieve según lo define Kimball, Reeves, Ross y Thornthwaite (2002). En la Tabla 2 se describe las dimensiones y hechos concernientes al proceso de negocio petitorio.

Tabla 2: *Descripción de Dimensiones y Hechos*

Dimensión	Descripción
D01 dim_cuadernillo	Contiene información concerniente al expediente judicial
D02 dim_recurso	Contiene información referente al recurso formulado en el petitorio: Casación, Apelación, Queja, etc.
D03 dim_especialidad	La especialidad a la cual corresponde el proceso: Civil, Constitucional, Penal, Laboral, Familia, entre otros.
D04 dim_materia	Clase a la que pertenece el proceso
D05 dim_organo_jurisdiccional	Corresponde a las Salas y juzgados que integran el organismo judicial según la ley orgánica del mismo, cada caso judicial abierto está vinculado a un órgano jurisdiccional.
D06 dim_tiempo	Contiene información referente al calendario esta permite analizar la información basándose en periodos de evaluación.
D07 dim_tema	Contiene información referente al pedido que realiza la parte procesal; no puede cambiar durante el tiempo de vida del proceso judicial.
D08 dim_norma	Corresponde a la parte normativa, fundamentos de ley, sobre la cual las partes procesales fundamentan su pedido.
F01 fact_petitorio	Contiene los hechos concerniente al proceso de negocio "Petitorio"

Fuente: Elaboración propia en base a Kimball, Reeves, Ross y Thornthwaite (2002)

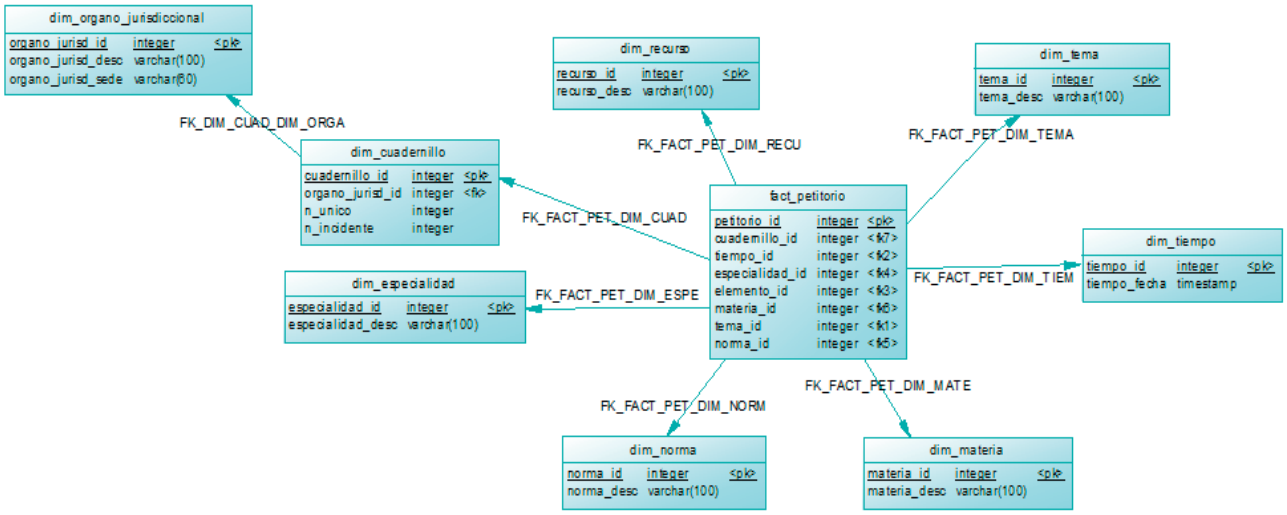

Figura 1: Modelo Dimensional Copo de Nieve
Fuente: Elaboracion Propia

En la figura 1 se presenta el diseño del modelo dimensional propuesto sobre el cual se circunscribe la aplicación y los resultados, la tabla fact_petitorio es la estructura que contendra a todos los hechos concernientes del proceso a evaluar, como se puede apreciar en la figura esta se interrelaciona con un conjunto de siete dimensiones; todo en conjunto permitirá formular el modelo de mineria de datos.

2.2 Diseño del Algoritmo MDD

Considerando el esquema distribuido que presenta el negocio en cuestión; una base de datos en cada sede judicial, se suma la sensibilidad y confidencialidad de la información que este tipo de organización maneja, infraestructura de tecnologías de información legadas; todo ello nos orienta al uso de minería de datos distribuida con lo cual se mantendrá la confidencialidad de la información al evitar su centralización, se reutilizará la infraestructura tecnológica y arquitectura de datos actual que presenta el negocio, se requiere mínimo presupuesto y se puede ver resultados a corto plazo.

La técnica de minería de datos elegida es K-means; esta técnica pertenece a la clase de técnicas de tipo clustering, esta permitirá agrupar los hechos del datamart propuesto según patrones de afinidad. El número de grupos o clusters resultantes ha obtener es configurable en nuestro caso definiremos cinco clusters inicialmente.

La técnica K-means presenta una serie de variantes de implementación; a continuación se formula una propuesta aplicable al contexto del presente estudio.

2.3 Algoritmo

La propuesta algorítmica consiste en el siguiente flujo:

1.- Sea $\Phi = \cup_{i=1}^{n} \beta_i \quad \forall\, i\, \varepsilon\, [1,n]$ el conjunto global de la información distribuida en sedes, donde cada β_i representa la base de datos de la sede i y n representa el número de bases de datos de todas las sedes judiciales.

2.- Sea μ el modelo dimensional propuesto el cual se considera desplegado en las n sedes.

3.- Se elige las sedes a incluir en la evaluación y se indicara el número de clusters a generar.

4.- Sea $\delta = \cup_{k=1}^{m} d_k \quad \forall\, k\, \varepsilon\, [1,m]$ el dataset resultante de la ejecución de μ en las sedes m seleccionadas; considérese $m <= n$ respectivamente asimismo las instancias de δ son datos cualitativos y nominales.

5.- De δ elegir c centroides aleatoriamente

6.- Determinar conglomerados considerando la distancia euclidiana de la instancia al centroide s (Ii → C_s) $\forall\, I_i\, \varepsilon\, \delta$; determinar la distancia de a I_i cada centroide considerándose que la instancia i pertenece al conglomerado si y solo si la distancia de la instancia al centroide es la mínima distancia determinada del conjunto de distancias calculadas para con respecto al conjunto de centroides.

$$I_i\, \varepsilon\, C_s \;\therefore\; distancia(I_i, C_s) = \min\{d_{i1}, d_{i2},, d_{ic}, \ldots d_{i(c-1)}, d_{ic}\}$$

7.- Repetir el paso 5 y paso 6 hasta que los nuevos centroides determinados en la iteración sean los mismos de la iteración $p-1$ con lo cual se podría concluir el proceso iterativo.

8.- Presentar la distribución final:

$$K_1 = \cup_{i=1}^{a_1} I_i \quad \forall\, i\, \varepsilon\, [1, a_1], \quad K_2 = \cup_{i=1}^{a_2} I_i \quad \forall\, i\, \varepsilon\, [1, a_2],$$
$$K_c = \cup_{i=1}^{a_m} I_i \quad \forall\, i\, \varepsilon\, [1, a_m]$$

Donde: $a_1 + a_2 + a_3 \ldots + a_m \leq n$, $s \leq c$ y K_c y son conjuntos de instancias disjuntos.

2.4 Arquitectura del Prototipo

La arquitectura del prototipo se enfoca en una arquitectura basada en tres capas, aplica el patrón arquitectonico Modelo Vista Controlador (MVC) Pressman (2010) en su diseño.

La figura 2 explica la arquitectura y su flujo se detalla a continuación:

El usuario inicia el proceso de clustering mediante el uso de un formulario web IU_Petitorio.jsp. Este evento es atendido por un Servlet: CTRL_Petitorio.

El servlet definido cumplirá el rol de controlador-delegador; este toma la petición procedente de la interfaz web y realiza las siguientes tareas:

1. Valida los datos consignados por el usuario como las sedes a intervenir en la evaluación, el número de clusters de interés.
2. Invoca al servicio web WsTdm la ejecución del método clustering.
3. Recibe el resultado del servicio web WsTdm siendo estos transferidos al usuario a través del formulario web: IU_Petitorio.jsp

La sección **Web Service WsTdm** (Elaboración propia) contiene parte de la especificación del servicio web; este realiza las siguientes acciones con la finalidad de atender el requerimiento del servlet:

1. Ejecuta el método: setData() con la finalidad de establecer las características de las fuentes de datos implicadas en la evaluación.
2. Ejecuta el método getData() con el objetivo de recuperar la información en un objeto Instancias.
3. Ejecuta el método clustering() pasándole como parámetro el objeto Instancias.
4. Retorna los clusters resultantes de la evaluación al servlet.

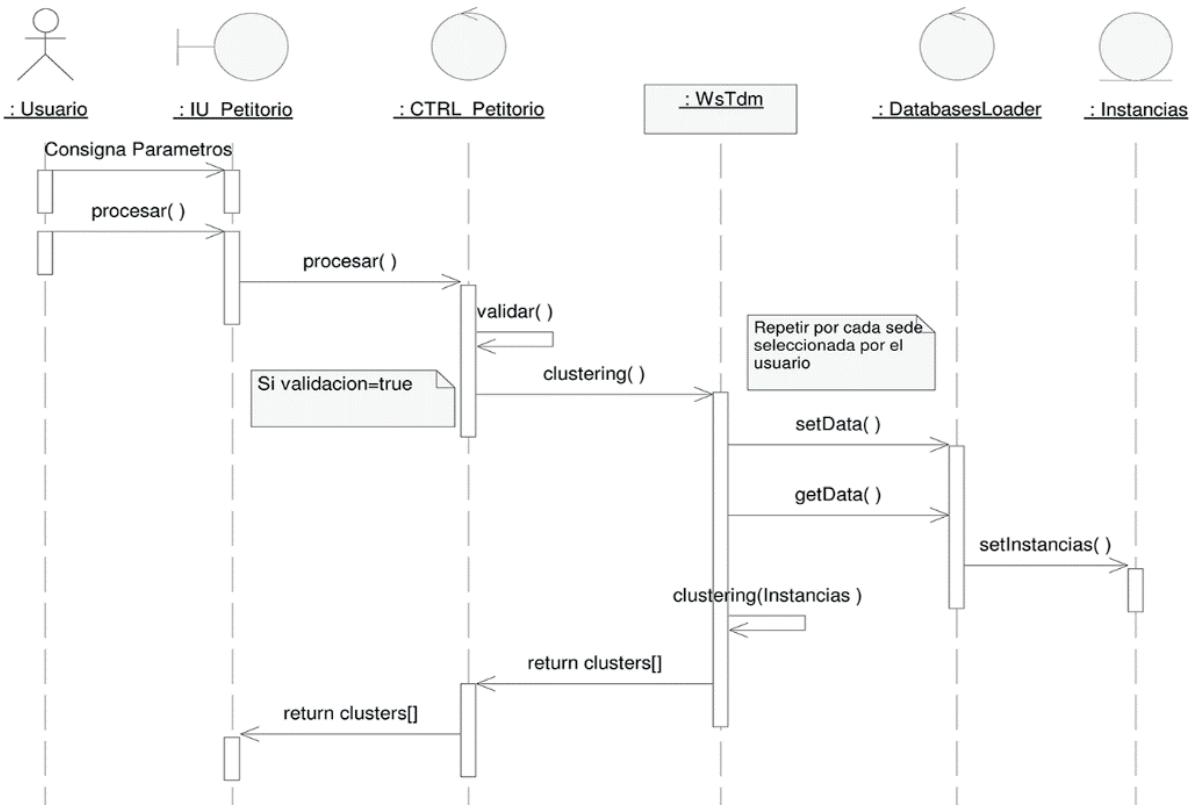

Figura 2: Arquitectura del Prototipo
Fuente: Elaboración propia

Web Service WsTdm

```
1.  package com.zemr.tesis;
2.  import java.util.ArrayList;
3.  import javax.jws.WebService;
4.  import javax.jws.WebMethod;
5.  import javax.jws.WebParam;
6.  import javax.xml.ws.Holder;
7.  import weka.clusterers.SimpleKMeans;
8.  import weka.core.Instances;
9.  import weka.core.Utils;
10. import weka.core.converters.DatabasesLoader;
11. import java.util.List;
12. @WebService(serviceName = "WsTdm")
13. public class WsTdm {
14.    @WebMethod(operationName = "clustering01")
15.      public List<miresultado> clustering01(
16.      @WebParam(name = "parameter1") int n_dataset,
17.      @WebParam(name = "parameter2") List<String> s_url,
18.      @WebParam(name = "parameter3") List<String> s_user,
19.      @WebParam(name = "parameter4") List<String> s_pwd,
20.      @WebParam(name = "parameter5") List<String> s_query,
21.      @WebParam(name = "parameter6") int n_clusters) {
22.         List<miresultado> result=new ArrayList(); double lb_error=0.0;
23.       try { DatabasesLoader nodo =new DatabasesLoader(n_dataset);
24.           for(int i=0;i<n_dataset;i++)
25. nodo.setSource(s_url.get(i),s_user.get(i),s_pwd.get(i),s_query.get(i));
26.           Instances instances=nodo.getData();SimpleKMeans s=new
    SimpleKMeans();s.setNumClusters(n_clusters);
27.           s.buildClusterer(instances);int i=s.getClusterCentroids().
    numInstances();int mclusterSizes[]=new int[i];
28. mclusterSizes=s.getClusterSizes();lb_error=s.getSquaredError();int[][]
    clusterOrden; clusterOrden= new int[mclusterSizes.length][mclusterSizes.
    length];clusterOrden=ordBurbuja(mclusterSizes);
29.           for(int v=0;v<clusterOrden.length;v++)
30.           {  miresultado unResult =new miresultado();
31.              unResult.setUid(v);unResult.setNombre("Cluster");
32. unResult.setPorcentaje((clusterOrden[v][1]*100.00)/(Utils.
    sum(mclusterSizes)*1.0));
33. unResult.setErrorCuadrado(lb_error);unResult.
    setInstancias(clusterOrden[v][1]);unResult.setDetalle(s.
    getClusterCentroids().instance(clusterOrden[v][0]).toString());result.
    add(unResult);
34.           }} catch (Exception e) {System.out.println("\n"+e.getMessage()); }
    return result; }}}
```

En **Servlet Ctrl_Petitorio** (Elaboración propia) se expone el código fuente del controlador, este objeto realiza la carga de las instancias de datos correspondiente a las n sedes definidas en la interfaz IU_Petitorio.jsp, asimismo es el encargado de invocar al servicio web, asignarle las instancias y recepcionar los resultados que este retorne convertirlo a formato JavaScript Object Notation (JSON) y remitirlo a la interfaz IU_Petitorio.jsp tal como se aprecia en la Figura 3.

Servlet Ctrl_Petitorio

```
1.  package servlet;
2.  import com.zemr.tesis.WsTdm_Service;
3.  import java.io.*;
4.  import java.util.ArrayList;
5.  import java.util.List;
6.  import javax.servlet.ServletException;
7.  import javax.servlet.http.HttpServlet;
8.  import javax.servlet.http.HttpServletRequest;
9.  import javax.servlet.http.HttpServletResponse;
10. import javax.servlet.RequestDispatcher;
11. import javax.xml.ws.WebServiceRef;
12. import com.zemr.tesis.Miresultado;
13. import org.json.simple.*;
14. public class CTRL_Petitorio extends HttpServlet {
15.    @WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost_8080/
    AppWsTdm/WsTdm.wsdl")
16.    private WsTdm_Service service;
17.    @Override
18.    protected void doPost(HttpServletRequest request, HttpServletResponse
    response)
19.          throws ServletException, IOException {
20.       int n_dataset=0,n_clusters=0,n_corte=4;
21.       String ls_csj;
22.       try {RequestDispatcher dispacher = request.
    getRequestDispatcher(request.getParameter("nombreJSP"));
23.          List<String> v_url=new ArrayList<String>();
24.          List<String> v_user=new ArrayList<String>();
25.          List<String> v_pwd=new ArrayList<String>();
26.          List<String> v_query=new ArrayList<String>();
27.          n_clusters=Integer.valueOf(request.getParameter("ncluster"));
28.          for(int i=1;i<=n_corte;i++)
29.          { ls_csj=String.valueOf(request.getParameter("CSJ0"+i));
30.             if (ls_csj.equals("null")==false)
31.             { v_query.add("call sp_fact_petitorio("+0+")");
32.               v_url.add("jdbc:sybase:Tds:server:2638/bd11_"+"0"+i);
33.               v_user.add("dba");v_pwd.add("sql");n_dataset++;}}
34.          com.zemr.tesis.WsTdm port = service.getWsTdmPort();
35.       java.util.List<Miresultado> v_resultado=new ArrayList<Miresultado>();
36. v_resultado=port.clustering01(n_dataset,v_url,v_user,v_pwd,v_query,n_
    clusters);
37.       request.setAttribute("n_dataset",String.valueOf(n_dataset));
38.          for(int i=0;i<n_clusters;i++){
39. request.setAttribute("r"+String.valueOf(i),v_resultado.get(i).getNombre()+
40.                String.valueOf(v_resultado.get(i).getUid()+1)+" Instancias="+
41.                String.valueOf(v_resultado.get(i).getInstancias())+" "+
42.                String.valueOf(v_resultado.get(i).getDetalle())
43.                );}
44.          resultado m_resultado=new resultado();
45. m_resultado.setErrorCuadrado(v_resultado.get(0).getErrorCuadrado());
46.          m_resultado.setMiResult(v_resultado);
47.          request.setAttribute("m_resultado",m_resultado);
48.          String json = null;
49.          JSONArray js = new JSONArray();
50.          JSONObject  j;
51.          for (int x = 0; x < v_resultado.size(); x++) {
52.             j = new JSONObject();
53.             j.put( "name",String.valueOf("Cluster "+(v_resultado.get(x).
    getUid()+1) ));
54.             j.put( "y",Math.round(v_resultado.get(x).getPorcentaje() * 100.0) /
    100.0);js.add(j);}json = js.toJSONString();
55.          request.setAttribute("json",String.valueOf(json));
56.          System.out.println(json);dispacher.forward(request, response);
57.       } catch (Exception e) {e.printStackTrace(); System.out.println("\n"+e.
    getMessage());}}}
```

2.5 Diseño del Prototipo

El prototipo de aplicación de minería de datos propuesto es una aplicación web desarrollada en lenguaje de programación java según la arquitectura de aplicación definida en la Figura 2.

En la Figura 3 se expone el diseño del prototipo el cual se explica a continuación.

El formulario se divide en tres secciones: la primera sección presenta el título del proyecto. La segunda sección contiene la lista parcial de sedes judiciales consideradas en el presente proyecto y los botones Procesar y Limpiar. La tercera sección contiene los resultados de la evaluación del clustering; Se tiene un ítem por cada cluster o conglomerado generado.

La estructura del ítem consiste de: un número y/o identificador del cluster, el total de instancias que lo conforman, el porcentaje que representa este con respecto al total de las instancias evaluadas y por último el detalle del conglomerado; las características que definen el conglomerado o cluster resultante.

En la parte superior del formulario se puede apreciar el grafico estadístico que refleja los resultados de la evaluación en cantidades porcentuales.

La sección **IU_Petitorio.jsp** (Elaboración propia) corresponde al código fuente que sostiene al formulario web de la figura 3, este código utiliza HTML

5 para el diseño del formulario web, hojas de estilo, librerías javascript, jquery.js y highcharts.js para brindar la presentación que se ofrece en el formulario web y los gráficos estadísticos que esta contiene.

IU_Petitorio.jsp

```
<%@page import="java.util.Iterator"%>
<%@page import="servlet.resultado"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <% resultado m_resultado =  (resultado)request.getAttribute("m_resultado");
        String n_dataset      = (String)request.getAttribute("n_dataset");
        String result="";  %>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Prototipo Aplicacion DDM</title>
        <link rel="shortcut icon" href="itinnovacionLogo.jpg">
        <link href="style03.css" rel="stylesheet">
        <script src="jquery-1.11.1.min.js"></script>
    <link rel="stylesheet" href="nivo-slider/themes/default/default.css"/>
        <link rel="stylesheet" href="nivo-slider/nivo-slider.css"/>
        <script type="text/javascript" src="nivo-slider/jquery.nivo.slider.js"></script>
        <script src="js/highcharts.js"></script>
        <script src="js/modules/exporting.js"></script>
        <style type="text/css">${demo.css}</style>
    <script type="text/javascript">
    $(function () { var datos = ${json};
    $('#estadistica1').highcharts({
    chart: {plotBackgroundColor: null, plotBorderWidth: 1,
            plotShadow: false},
    title: {text: 'Resultados porcentuales del Clustering del Petitorio'},
            tooltip: {pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'},
    plotOptions: { pie: {allowPointSelect: true, cursor: 'pointer', dataLabels: {enabled: true, format: '<b>{point.name}</b>: {point.percentage:.1f} %',
    style: { color: (Highcharts.theme && Highcharts.theme.contrastTextColor) || 'black'}}}},
    series: [{type: 'pie', name: '% del Cluster', data: datos}]});});
    </script> </head> <body>
    <div id="topContent">  <div id="estadistica1"> </div>
    <h1 id="logo"><a href="#">Prototipo Aplicación de Minería de Datos</a></h1></div><!-- topContent --><div id="container">
    <div id="cabecera"><form action="CTRL_Petitorio" method="POST">
    <input type="hidden" name="nombreJSP" value="/IU_Petitorio.jsp" />
    <table class="sedes"><tr><th><h4>Distritos Judiciales:</h4></th>
    </tr><tr><td><input type="checkbox" name="CSJ01" value="1" checked>Corte Suprema<br></td><td>
    <input type="checkbox" name="CSJ02" value="2">Corte Lima Norte<br></td></tr><tr><td><input type="checkbox" name="CSJ03" value="3">Corte Lima Sur<br>
    </td><td><input type="checkbox" name="CSJ04" value="4">Corte Lima Este<br></td></tr><tr><td>N° de Clusters: <input type="number" name="ncluster" value="5" min="2" max="150" step="1" size="2"><br></td></tr></table><div id="tarea"><td> </td>
                <input type="submit" class="button_1" name="procesar" value="Procesar"><input type="reset" class="button_1" name="reset" value="Limpiar"></div></form>
            </div><!-- cabecera --><div id="detalle"><h4></h4><%
                if (m_resultado != null && m_resultado.getMiResult().size()>0){out.println("<h4><b>Resultados de la Evaluación:</b></h4>");out.println("<td><b>Error Cuadrático: "+String.valueOf(m_resultado.getErrorCuadrado())+"</b></td>");}%>
            <table class="tProductos">
    <%if (m_resultado != null && m_resultado.getMiResult().size()>0)  {
                    out.print("<td><b>N°Cluster</b></td>");
                    out.print("<td><b>Instancias</b></td>");
                    out.print("<td><b>Instancias(%)</b></td>");
                    out.print("<td><b>Caracteristicas del Cluster</b></td>");
                for(int i=0;i<m_resultado.getMiResult().size();i++){
                    out.println("<tr>");
    result=String.valueOf(m_resultado.getMiResult().get(i).getUid() + 1);
                    out.println("<td>" + result + "</td>");
    result=String.valueOf(m_resultado.getMiResult().get(i).getInstancias());
                    out.println("<td>" + result + "</td>");
    result=String.valueOf(Math.round(m_resultado.getMiResult().get(i).getPorcentaje()*100.0)/100.0);out.println("<td>" + result + "</td>");
    result=String.valueOf(m_resultado.getMiResult().get(i).getDetalle());
                    out.println("<td>" + result + "</td>");
                    out.println("</tr>");}}%> </table </div></div><!-- container --></body>
</html>
```

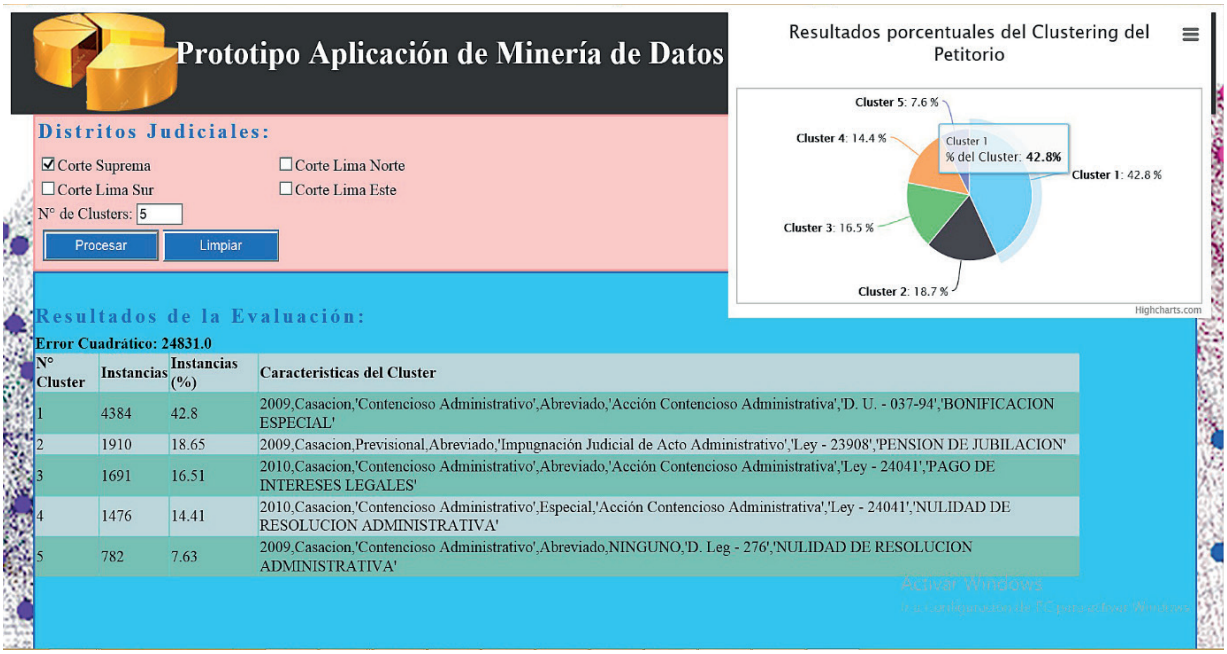

Figura 3: Prototipo de la Aplicación

Fuente: Elaboración propia

3. RESULTADOS Y DISCUSIÓN

Los resultados obtenidos nos permitirán fundamentar los objetivos del presente trabajo. La muestra poblacional de análisis corresponde a un órgano jurisdiccional casatorio del periodo 2008 al 2010, la figura 4 grafica la distribución porcentual de los resultados obtenidos los cuales se analizan a continuación:

La interpretación de los conglomerados resultantes nos llevan a determinar que el 42.8% de la carga procesal corresponden a procesos casatorios iniciados en el año 2009 correspondiente a la subespecialidad contencioso administrativo en la cual la parte demandante solicita una bonificación especial fundamentando su pedido en el Decreto de Urgencia D.U 037-94.

El segundo conglomerado explica que se tiene un 18.65% de casos iniciados en el año 2009 solicitan Pensión de Jubilación y se fundamentan en la ley 23908 del código procesal civil; este pedido lo realizan mediante una impugnación judicial de acto administrativo.

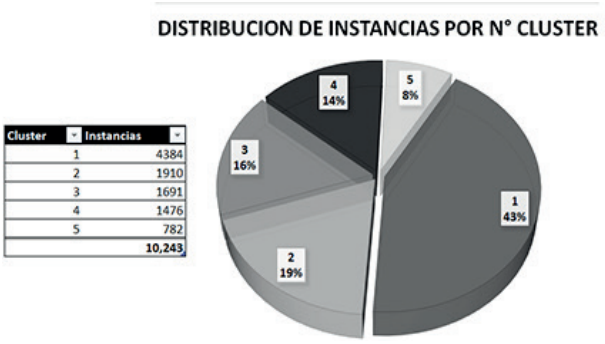

Figura 4: Distribución porcentual de la Evaluación
Fuente: Elaboración propia

El tercer conglomerado que equivale al 16.51% de la carga evaluada se trata de un pedido de Pago de Intereses Legales sosteniendo su pedido en la ley 24041, esta formulación se realiza mediante un proceso Abreviado con acción contencioso administrativo.

El cuarto conglomerado recae en un 14.41% quienes solicitan Nulidad de Resolución Administrativa respaldando su pedido mediante ley 24041, este petitorio lo formulan mediante un proceso especial de acto administrativo.

El quinto conglomerado presenta el porcentaje más bajo, el 7.63% de la carga procesal tiene como requerimiento una Nulidad de Resolución Administrativa y lo realizan mediante un proceso abreviado de tipo acción contencioso administrativa, se fundamenta el petitorio en el Decreto Ley 276 del Código Civil.

4. CONCLUSIONES

Se ha propuesto un algoritmo de clustering distribuido adaptable a la entidad judicial por su naturaleza organizacional: esquemas de negocios dispersos físicamente, confidencialidad de la información, reúso de infraestructura tecnológica, complejidad organizacional, bajo presupuesto.

Se diseñó un modelo multidimensional que ha posibilitado organizar la información experimental basada en data nominal del proceso petitorio.

Se desarrolló un prototipo de software el cual ha permitido aplicar el algoritmo propuesto, el modelo multidimensional planteado y el uso de tecnologías *Ad hoc* a fin de predecir patrones de comportamiento en el petitorio de las partes procesales y exponer los resultados de la presente investigación.

La implementación de la presente propuesta en la organización del presente caso de estudio apoyará en el cumplimiento de los objetivos señalados en la agenda estratégica institucional como en el fortalecimiento de mecanismos para la reducción del volumen procesal, plazos procesales y nivel de litigiosidad; con lo cual se logrará una mejora en la calidad de los servicios que esta brinda a los ciudadanos.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] Breña, W. (2008). *La carga procesal bajo la lupa por materia y tipo de organo jurisdiccional*. Justicia Viva, Volumen (1)

[2] Hurtado, F. (2005). *Segmentación de clientes usando el algoritmo de clustering K-Mean*. (tesina de pregrado). UNMSM, Lima

[3] Kimball, R., Reeves, L.,Ross, M. y Thornthwaite, W. (2002). *The Data Warehouse Lifecycle Toolkit.* USA: Wiley

[4] Mendoza, E. (2013). *Agenda Estrategica del Poder Judicial*. Recuperado de https://www.pj.gob.pe

[5] PMSJ, (2014). *Proyecto de Mejoramiento de los Servicios de Justicia*. Recuperado de http://www.pmsj.org.pe

[6] Pressman, R. (7 Ed.) (2010). *Ingenieria de Software Un enfoque Practico.* (pp. 215-233) Mexico: Mc Graw Hill

[7] Rekha, S. y Sabu, M. (2010). *Survey on Distributed Data Mining in P2P Networks*. Recuperado de https://arxiv.org

[8] SGPPCM (2013). *Política Nacional de Modernización de la Gestión Pública al 2021*. Recuperado de http://www.pcm.gob.pe

[9] Veramendi, E. (2008). *El petitorio implícito en los procesos de familia: A propósito del tercer pleno casatorio*. Recuperado de http://boletinderecho.upsjb.edu.pe